ILS2014
5th International Conference
Information Systems, Logistics and Supply Chain
August 24-27 2014
Dinalog | Beta | TRAIL | Breda, The Netherlands

Proceedings of the 5th International Conference on Information Systems, Logistics and Supply Chain CONNECTING WORLDS ILS 2014 – Breda (Netherlands), August 24-27

Minimizing work overload in mixed model assembly lines: A case study from truck industry

K. Aroui[1,*], G.Alpan[1], Y. Frein[1]

[1]Grenoble INP/ UJF-Grenoble 1/ CNRS, G-SCOP UMR5272 – France

karim.aroui@grenoble-inp.fr

Abstract: *In this article, we consider the mixed model assembly line sequencing problem with work overload minimization (MMSP-W). We consider an assembly line having operators with different characteristics. We propose a formulation based on a linear programming approach. The model is implemented using the CPLEX solver. Numerical tests are carried out on generated data and on an industrial case study from a truck assembly line. Computational results demonstrate the considerable improvements obtained in comparison to the traditional car sequencing approach.*

Keywords: *Mixed model assembly line, sequencing, work overload, linear programming*

1 Introduction

The need to manufacture a wide variety of products has led to many changes in the requirements of production systems. Originally, using the single model assembly lines, manufacturing companies have efficiently produced large quantities of the same product. But the diversification of customer demand necessitated the invention of more modern manufacturing methods. Mixed Model Assembly line (MMAL) is a production line where different variants of common products (called models) are intermixed to be assembled on the same line (Monden, 1983). This type of assembly line is used in Just-In-Time (JIT) systems and applied in a wide range of industries: from electronic assembly to automotive industry. It helps matching production to changing customer requirements while keeping small stock sizes. In MMAL, products move on a paced conveyor belt and operators move on the conveyor while performing their tasks on products.
Line balancing and product sequencing are the two main problems in MMAL that attract the attention of researchers. The line balancing problem deals with the assignment of tasks to workstations. The most common objective for this problem (Scholl, 1995) is to minimize the number of workstations for a given cycle time (i.e., production rate).

The sequencing problem assumes that the line has already been balanced. The main task is to find the production sequence of a given number of models within a given planning horizon, e.g., one day or shift (Boysen et al., 2009). Some recent works also consider the sequencing and line balancing problems simultaneously (see, for instance, Mosadegh et al., 2012).
This paper deals with the mixed model assembly line sequencing problem. In the literature, two basic objectives are considered to solve this problem: the constant rate of part usage (Monden, 1983 ; Miltenburg, 1989 ; Bautista et al., 1996 ; Boysen et al., 2009), and the leveling of work load (Yano et Bolat, 1989 ; Xiabo et Ohno, 1997 ; Bautista et Cano, 2008). Some authors consider both objectives simultaneously (Aigbedo and Monden, 1997; Kotani et al., 2004).

Boysen et al. (2009) records three different sequencing approaches to attain these objectives:

– Mixed-model sequencing: the purpose of this approach is to minimize the work overload (or maximize the total work performed) by taking into account the processing times of each product on every workstation.

– Car sequencing: also seeks to minimize sequence-dependent work overload, but in an implicit manner, by using sequencing rules: limited frequency at which special options may appear.

– Level scheduling: This approach aims to have regular consumption rates of materials.

Car sequencing approach is the most common approach in the literature and extensively applied in automotive industry. In spite of its popularity, this approach has some drawbacks: first of all sequencing rules are not trivial to construct. And secondly the respect of these rules may lead to unexpected work overload on the assembly line because they fail to capture the real work overload to minimize (see Lesert et. al., 2011). In this article, we choose to act directly on the work overload. Hence, the Mixed-Model Sequencing Problem with Work overloads Minimization (MMSP-W) is considered. This approach is poorly developed in the literature. Nevertheless, we find some works by Yano and Rachamadugu (1991), Bolat and Yano (1992a, 1992b), Scholl et al. (1998), Xiaobo and Ohno (1997, 2000), Bautista and Cano (2008, 2011) and Bautista et al. (2012), among others.

Yano and Rachamadugu (1991) consider MMSP-W with one workstation and two product types and propose an approach leading to the optimal solution when certain conditions are met. Then, they generalize the approach to *k* workstations and two product types. Bolat and Yano (1992a) introduce a surrogate objective for utility work at a single station assembly line. Optimal solution can be obtained depending on certain conditions on the parameter values.

Still for the case of one workstation and two product types, Bolat and Yano (1992b) develop three optimal solution procedures and a heuristic to minimize the work overload.

Exact methods, such as Branch and Bound have been used (Xiaobo and Ohno, 1997). However, the computation times are acceptable only for small instances. Heuristics are also used. For instance, Scholl et al. (1998) solve the MMSP-W using an informed tabu search procedure with a pattern based vocabulary building strategy.

Bautista and Cano (2011) and Bautista et al. (2012) propose models for the MMSP-W for assembly lines considering workstation dependencies (operations in a workstation could begin only if the previous station finished his operations). The authors propose a heuristic based on a Bounded Dynamic Programming approach. The method combines some characteristics of dynamic programming (determining the shortest path in a graph) with Branch and Bound (searching a bound of the optimal solution).

This work is based on an industrial case study of one of the plants of Volvo Group Trucks Operations. The models with only two types of products are totally ineffective in this industrial context. Furthermore, the study of the industrial case reveals three types of operators with different specifications. Two of them were never modeled in the literature. Our study proposes a model considering the three existing types of operators.

The truck assembly lines are typical MMAL with highly diversifying products. Compared to the automobile industry, the number of vehicles to produce per day is much lower, which may be an opportunity to test exact methods in modeling.

This article is organized as follows: Section 2 introduces fundamental assumptions and describes the problem. In section 3, we propose a linear programming model for the MMSP-W. Section 4 describes the numerical experiments based on an industrial case study. The paper ends with concluding remarks.

2 Problem description

In the case of a paced assembly line, the products move on the line with a constant speed. The time between two consecutive products is constant and is called the cycle time. The operator follows the product with the same speed as the pace of the assembly line (meters/time unit) while doing his tasks. Therefore, the cycle time is oftentimes used to define the boundaries of a workstation as well (i.e. a workstation is a space characterized by the number of meters that the operator can make within the cycle time). We will use the term “time window” to refer to the available time for an operator to perform his tasks on a product. The work overload appears when the operator cannot finish the required tasks on a product within the predefined time window. The work overload refers to the remaining work.

There are several ways to handle this remaining work: For instance, in Bautista et al. (2012), the excess work is considered as unfinished work and is handled at the end of the line. In Bautista and Cano (2011) and in Boysen et al. (2011), the productive activity is increased above the standard, using the assistance of reinforcement operators. In Xiaobo and Ohno (1997), the assembly line is stopped to complete the pending work.

The characteristics of the assembly line considered in this study and the assumptions regarding the work overload are given below:

– the assembly line balancing is already done. All tasks are assigned to the operators.
– the assembly line is a paced assembly line.
– the processing times are assumed to be deterministic. Set-up times are included in processing times and the time required to move from the downstream to the upstream station limits are negligible.
– the operators cannot go beyond the limits of their time window (due to equipment restrictions) and there are no buffers between stations. This means that, no extra work should remain from a preceding operator. Thus, there are no dependencies between the consecutive operators. In case of work overload, the operator has to rush or reinforcement operators assist him to finish the job on time;
– a workstation can have several operators. They can perform their tasks independently.

When the operator has to rush to finish his operations on time, the risk of defects, accidents due to fatigue or health problems due to bad postures, etc. increase. The use of reinforcement operators to handle the work overloads is expensive. Our objective is to generate product sequences to minimize the work overload. The proposed method is intended to be used in an industrial context.

The following example illustrates those characteristics and assumptions: sequencing 3 products for 2 consecutive operators. Operation times for each product are given in table 1. The cycle time is 5 time units.

Product	*m1*	*m2*	*m3*
operator 1	5	6	3
operator 2	6	4	4

Table 1: Operation times for the example

Figure 1 shows the Gantt diagram of the sequence *m2*, *m1* then *m3*. We can see that the work overload generated by product *m2* for the operator 1 impacts the next product in the sequence (that is, *m1* which generates further delay). When operator 2 finishes his operations on *m2*, he can start jobs on *m1* only when it is inside his workstation. In this case, an idle time occurs after processing *m2*.
The operators are independent so the delay observed on *m2* for the operator 1 will not impact the beginning of operation of operator 2 on *m2*.

We note that, in practice, when the operator 1 rushes to finish the tasks on *m2* or when the reinforcement operators are used, this delay will not be observable on the product *m1*. Even though such delays are not "observable in practice", if we can avoid them we also avoid the risks generated by a rushing operator or the cost of a reinforcement operators. Therefore, in the model described in section 3, we focus on the minimization of this type of delays.
The optimization criterion considered in this study is hence minimization of total work overload (sum of gray squares in Figure 1).

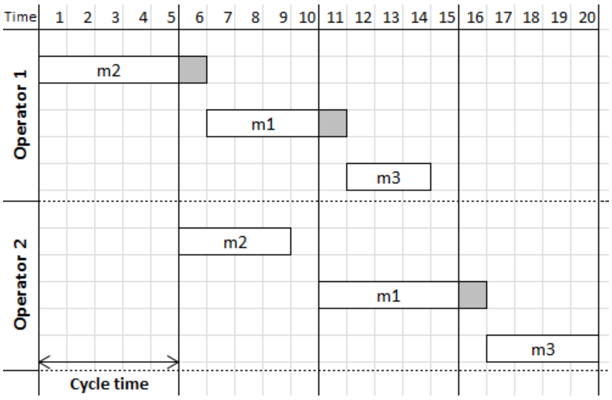

Figure 1: Dynamics of work overload depending on the product sequence

The proposed model in section 3 considers only the overloads. When an idle time occurs, it will not be taken into account in the criterion.

3 A model based on mixed integer linear programming

3.1 Operator types

Three types of operators exist in our case study:

– type 1 operators: are regular operators and they are assigned tasks to be performed on every vehicle within one cycle time. Time window for type 1 operators is one cycle time for all vehicles.

– type 2 operators: the tasks given to these workers are required only by some vehicles (special options). They are allowed to work in a known number of cycle times. This number can be different for each vehicle of the sequence. Time window for type 2 operators depends on vehicles.

– type 3 operators: in the balancing problem, some tasks, that must be executed on all vehicles, could not be split and require more than a single cycle time to be performed. These tasks are assigned to operators of type 3. The operators perform their tasks on a known number of cycle times. This number is the same for all products that the operator has to do. An operator of type 3 has to be associated to one or more operators of the same type with whom he has to alternate products depending on their positions in the sequence. For example, if the tasks are balanced on two cycle times, we will have two operators. One will operate on vehicles in even positions and the other on vehicles in odd positions in the sequence. Each operator will have a time window of two cycle times.

As shown in Figure 2 an operator can work on several cycle times. In the example the line contains 7 workstations and 10 operators. Operators 1 to 3, 6 and 8 to 10 are of type 1. Each of them has a time window of one cycle time. Operator 7 is of type 2. His time window depends on vehicles (up to three cycle times). And operators 4 and 5 are two coupled operators of type 3. Each of them takes every other vehicle on the sequence and has a time window of two cycle times.

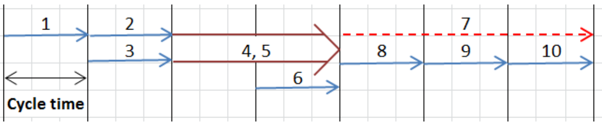

Figure 2: Example of assembly line

We propose a model using linear programming approach to solve the mixed-model sequencing problem with work overload minimization which takes into account the assumptions considered in section 2 and the characteristics of each operator type to suit the condition of the case study.

The parameters and variables of this model are presented below.

Parameters

n	Number of products
P	Number of operators
P_1	Set of operators of type 1
P_2	Set of operators of type 2
P_3	Set of operators of type 3
i	Product index
j	Position index
p	Operators index
γ	Cycle time
t_{ip}	Processing time required by product i for operator p.
d_{ip}	For operators of type 1 and 2, $d_{ip} = t_{ip} - \gamma$. For operators of type 3, $d_{ip} = t_{ip} - nc_p * \gamma$
$\propto_{ip}$	Binary variable that is equal to 1 if $t_{ip} \neq 0$ and 0 otherwise ($p \in P_2$)
np_{ip}	Number of cycle times where an operator p of type 2 is allowed to work for the product i ($p \in P_2$)
a_{jp}	Binary variable that is equal to 1 if the operator p of type 3 performs tasks on the product of the position j, and 0 otherwise ($p \in P_3$)
nc_p	Number of cycle times for operator p of type 3 ($p \in P_3$)
M	Big integer

Variables

x_{ij}	Binary variable that is equal to 1 if a product *i* is assigned to position *j* of the sequence, and 0 otherwise
c_{jp}	Delay or idle time for operator p for the product on position j of the sequence
r_{jp}	work overload for operator p of type 1 for the product on the position j
b_{jp}	Intermediary variable to calculate work overload for type 2 operators
w_{jp}	Work overload for operator p for the product on position j of the sequence

Figure 3 shows an example of an operator of type 2 and a sequence of 8 products. The operator has no tasks to perform on products m2, m3, m5, m7 and m8 (α_{ip} =0) so they don't appear in the Gantt chart. Overloads appear when the worker reaches the limit imposed by the time window ($np_{ip} * \gamma$). They are characterized by the gray squares in figure 3.

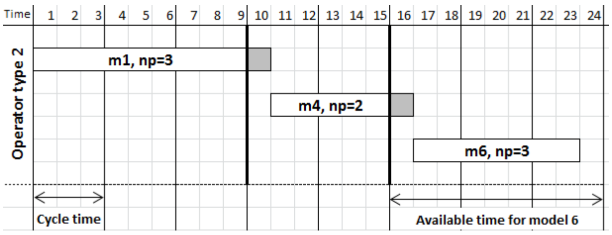

Figure 3: Gantt char of an example of operator of type 2

Figure 4 illustrates a Gantt chart of an example of sequencing 7 products on three linked operators of type 3.
Time window for each operator and each product is $nc_p * \gamma = 3 * 3 = 9$. Unlike operators of type 2, time window does not depend on products. All products will have the same available time to be performed by an operator of type 3.

Since the work is balanced on 3 cycle times, operator 1 will perform tasks on m1, m4 and m7. m2 and m5 will be treated by operator 2. m3 and m6 will be done by operator 3.

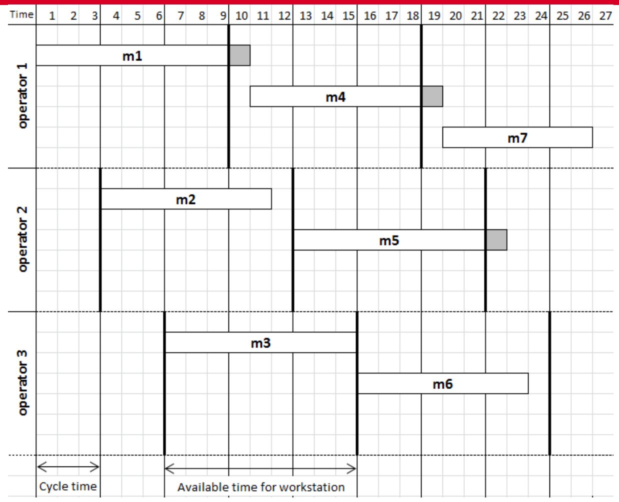

Figure 4: Gantt char of an example of type 3 operators

We will now present a mixed linear integer program to minimize the total work overload considering the specificities of each type of operator.

$$\mathrm{Min} \sum_{p=1}^{P} \sum_{j=1}^{n} \mathrm{w_{jp}}$$

subject to :

$$c_{jp} = r_{j-1,p} + \sum_{i=1}^{n} d_{ip} x_{ij} \quad \forall j \ \forall p \qquad (1)$$

$$r_{jp} \geq 0 \quad \forall j \ \forall p \qquad (2)$$

$$r_{jp} \geq c_{jp} \quad \forall j \ \forall p \qquad (3)$$

-- work overload for type 1 operators –

$$w_{jp} = r_{jp} \quad \forall j \ \forall p \in P_1 \qquad (4)$$

-- work overload for type 2 operators --

$$b_{jp} = \left(\sum_{i=1}^{n} np_{ip} \, x_{ij} - 1 \right) * \gamma \quad \forall j \forall p \in P_2 \qquad (5)$$

$$w_{jp} \geq 0 \quad \forall j \ \forall p \in P_2 \qquad (6)$$

$$w_{jp} \geq r_{jp} - b_{jp} - \left(1 - \sum_{i=1}^{n} \propto_{ip} x_{ij} \right) * M \quad \forall j \forall p \in P_2 \qquad (7)$$

-- work overload for type 3 operators --

$$w_{jp} \geq 0 \quad \forall j \ \forall p \in P_3 \qquad (8)$$

$$w_{jp} \geq \left(w_{j-nc_p,p} + \sum_{i=1}^{n} x_{ij} d_{ip} \right) * a_{jp} \quad \forall j \ \forall p \in P_3 \qquad (9)$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i \qquad (10)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \qquad (11)$$

$$x_{ij} \in \{0,1\} \quad \forall i \ \forall j \qquad (12)$$

$$c_{jp}, r_{jp}, b_{jp}, rc_{jp}, rt_{jp} \in \mathbb{R} \quad \forall j \ \forall p \qquad (13)$$

This model aims to minimize the total work overload; constraint (1) establishes that the delay or idle time of the product positioned in j is his delay (or idle time) in addition to the delay (or idle time) of the position j-1; constraints (2) and (3) indicates that $r_{jp} = max(0, c_{jp})$ since the objective function takes into account only the delays (and not idle times).

Constraint (4) specifies work overloads for type 1 operators.

Constraints (5)-(7) calculate work overloads for type 2 operators.
Let's note t_{jp} the processing time of the vehicle in position j: $t_{jp} = \sum_{i=1}^{n} t_{ip} x_{ij}$ and $\mathrm{np_{jp}}$ the number of cycle times where operator *p* of type 2 is allowed to work for of the vehicle in position j: $np_{jp} = \sum_{i=1}^{n} np_{ip} x_{ij}$. Obviously, work overload does not occur for products if they are not treated by an operator. Thus, the work overload for type 2 operators is:

$$w_{jp} = \begin{cases} \max\left(\mathrm{r_{jp}} - \left(\mathrm{np_{jp}} - 1\right) * \gamma \ , \ 0\right) & \text{if } \mathrm{t_{jp}} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Constraints (8) and (9) calculate work overloads for type 3 operators.
Let's note d_{jp} the delay or idle time of the vehicle in position j: $d_{jp} = \sum_{i=1}^{n} d_{ip} x_{ij}$. For operators of type 3, the work overload is:

$$w_{jp} = \begin{cases} \max\left(0, \mathrm{w}_{\mathrm{j}-nc_p} + \mathrm{d_{jp}}\right) & \text{if } \mathrm{a_{jp}} = 1 \\ 0 & \text{otherwise} \end{cases}$$

It's the work overload of the previous performed product (which is $\mathrm{j} - nc_p$ for operators of type 3) added to the delay or idle time of the vehicle in position j. Similar to type 2 operators, no work overload is considered if the worker has no tasks to do on the product.

Constraint (10) guarantees that only one position can be assigned to each product; constraint (11) indicates that only one product can be assigned in each position of the sequence; and, finally, constraint (12) requires that the assigned variables are binary.
Table 2 illustrates variables' values for operator 1 in the example presented in section 2.

	Sequence	d_j	c_j	r_j
Operator 1	*m1*	1	1	1
	m2	0	1	1
	m3	-2	-1	0
Operator 2	*m1*	-1	-1	0
	m2	1	1	1
	m3	-1	0	0

Table 2: Variables' values for the example

Table 3 explains variables' values assigned to type 2 operators in the example of Figure 2.

Sequence	t_j	np_j	α_j	r_j	b_j	w_j
m1	10	3	1	7	6	1
m2	0	0	0	4	-3	0
m3	0	0	0	1	-3	0
m4	6	2	1	4	3	1
m5	0	0	0	1	-3	0
m6	7	3	1	5	6	0
m7	0	0	0	2	-3	0
m8	0	0	0	0	-3	0

Table 3: Variables' values for the example of type 2 operators

Table 4 illustrates variables' values assigned to the first type 3 operator in the example illustrated in Figure 3.

Sequence	t_j	a_j	d_j	w_j
m1	10	1	1	1
m2	8	0	-1	0
m3	9	0	0	0
m4	9	1	0	1
m5	10	0	1	0
m6	8	0	-1	0
m7	7	1	-2	0

Table 4: Variables' values for the example of type 3 operators

4 Numerical experiments

We apply the linear program proposed in section 3 on two experiments: the first is a small academic instance taking into account all operator types and the second is our case study of the Volvo plant.
In both experiments, the solutions are obtained using the solver IBM ILOG CPLEX Optimization Studio V12.4 running on a Dell PC with Intel Core i5 2.50 GHz with 4GB of RAM using Windows 7 Enterprise.

Products	*Type 1 operators*					*Type 2 operators*		*Type 3 operators*		
	w1	*w2*	*w3*	*w4*	*w5*	*w6*	*w7*	*w8*	*w9*	*w10*
m1	7,24	8,59	6,74	5	8,8	20 (3)	13,5 (2)	22	22	22
m2	5,84	5,86	7,07	5,28	7,42	0	0	20	20	20
m3	5,84	5,86	5,53	5,28	4,74	14,3 (2)	19 (3)	21	21	21
m4	6,54	6,04	6,74	5	8,8	0	0	22	22	22
m5	5,84	6,19	6,82	5,49	4,74	0	0	19	19	19
m6	5,84	5,86	5,53	5,28	4,74	27,5 (4)	19 (3)	23	23	23
m7	7,24	6,97	5,75	7,29	4,74	0	0	20	20	20
m8	5,84	6,19	6,74	5	8,8	0	0	22	22	22
m9	5,84	5,86	6,41	5,28	4,74	0	0	19	19	19
m10	5,84	6,04	7,36	5	7,42	0	0	19	19	19
m11	7,24	8,74	6,74	5	8,8	0	13,5 (2)	21	21	21
m12	5,84	7,15	6,41	5,49	4,74	14,3 (2)	0	21	21	21

Table 5: processing times in minutes for the academic instance

Positions	Products	Type 1 operators					Type 2 operators		Type 3 operators		
		w1	*w2*	*w3*	*w4*	*w5*	*w6*	*w7*	*w8*	*w9*	*w10*
1	*m8*	0	0	0	0	1,8	0	0	1	0	0
2	*m6*	0	0	0	0	0	0	0	0	2	0
3	*m2*	0	0	0,07	0	0,42	0	0	0	0	0
4	*m7*	0,24	0	0	0,29	0	0	0	0	0	0
5	*m10*	0	0	0,36	0	0,42	0	0	0	0	0
6	*m12*	0	0,15	0	0	0	0,3	0	0	0	0
7	*m11*	0,24	1,89	0	0	1,8	0	0	0	0	0
8	*m9*	0	0,75	0	0	0	0	0	0	0	0
9	*m3*	0	0	0	0	0	0,3	0	0	0	0
10	*m4*	0	0	0	0	1,8	0	0	1	0	0
11	*m5*	0	0	0	0	0	0	0	0	0	0
12	*m1*	0,24	1,59	0	0	1,8	0	0	0	0	1

Table 6: work overloads for the solution of the academic instance

In-stances	W CPLEX	W Current Procedure	Improvement	Type 1 operators	Type 2 operators
				Improvement	Improvement
1	*917,34*	*2729,5*	*66,4%*	*39,7%*	*85,6%*
2	*979,52*	*3600,18*	*72,8%*	*42,5%*	*82,1%*
3	*895,7*	*1873,85*	*52,2%*	*27,9%*	*72,5%*
4	*1320,19*	*3295,43*	*59,9%*	*32,5%*	*77,3%*
5	*1947,28*	*3147,05*	*38,1%*	*33,8%*	*60,0%*
6	*942,3*	*1806,03*	*47,8%*	*32,9%*	*65,5%*
7	*783,97*	*1353,65*	*42,1%*	*32,6%*	*69,7%*
8	*1433,82*	*2243,03*	*36,1%*	*31,3%*	*63,6%*
9	*705,4*	*1263,83*	*44,2%*	*40,8%*	*64,6%*

Table 7: Results for the case study instances

Instances	W Current Procedure	*CPLEX 1 hour*		*CPLEX 2 hours*		*CPLEX 3 hours*	
		W	Improvement	*W*	Improvement	W	Improvement
1	*2729,5*	*1429,44*	*47,6%*	*1429,44*	*47,6%*	*917,34*	*66,4%*
2	*3600,18*	*1594,89*	*55,7%*	*990,51*	*72,5%*	*979,52*	*72,8%*
3	*1873,85*	*1296,36*	*30,8%*	*914,45*	*51,2%*	*895,7*	*52,2%*
4	*3295,43*	*1845,75*	*44,0%*	*1402,53*	*57,4%*	*1320,19*	*59,9%*
5	*3147,05*	*2433,24*	*22,7%*	*2433,24*	*22,7%*	*1947,28*	*38,1%*
6	*1806,03*	*1598,39*	*11,5%*	*972,5*	*46,2%*	*942,3*	*47,8%*
7	*1353,65*	*1085*	*19,8%*	*809,91*	*40,2%*	*783,97*	*42,1%*
8	*2243,03*	*1667,12*	*25,7%*	*1667,12*	*25,7%*	*1433,82*	*36,1%*
9	*1263,83*	*1060,67*	*16,1%*	*803,28*	*36,4%*	*705,4*	*44,2%*

Table 8: Improvements at each hour of execution for the case study instances

4.1 Numerical experiment with academic instance

The academic instance is an example with 12 products and 10 operators: 5 operators of type 1, 2 operators of type 2 and 3 operators of type 3. Table 5 gives the processing times by product and operator. The cycle time is 7 min. For operators of type 2, $\mathrm{np}_{\mathrm{ip}}$ is given in parenthesis next to the processing times. For operators of type 3, the available time to perform tasks on products is three cycle times ($nc_p = 3$).
The optimal solution is obtained in 1.76 seconds. The values of work overloads are given in table 6.

4.2 Numerical experiment with case study instances

We apply the linear program on nine instances from the industrial case study. Each one represents the demand plan of a production day. That is equivalent to 60 vehicles per day assembled by 92 operators: 77 of type 1, 12 of type 2 and 3 of type 3. The operators of type 3 have no variation in their processing times. There is no utility to consider them in our tests. As a result, 89 operators are considered. The cycle time is 7 minutes.

We limited the CPU time to 10800s (3 hours). A comparison between the solution given by CPLEX and the solution of the current sequencing method used in the production site is given in table 7. The current procedure is based on “manual” car sequencing approach (see section 1).

From tables 7 and 8, we can comment on the following observations:

- CPLEX did not obtain the optimal solution for all instances after three hours of execution. We could not guarantee an optimum solution but we improve significantly the solution given by the current procedure in all instances.
- The average improvement on total work overload W is 51%.
- After one hour of execution, the average improvement on W is 30,4%
- After two hour of execution, the average improvement on W is 44,4%

5 Conclusion

In this article, we present a linear programming model for sequencing products to minimize the work overload in a mixed model assembly line under three different types of worker operating conditions.
We perform numerical experiments with an academic instance and then with nine instances from an industrial case study where each instance represents a working day. The results of these tests demonstrate an average decrease of 51% in total work overload with the proposed solution, compared to the current procedure used in the company. We note that the comparisons are made for results obtained after 3 hours of execution of the linear program. We are, hence, unable to guarantee the optimality of these solutions.

Future research will focus on using dynamic programming and designing heuristic resolution procedures to solve the problem.

References

Aigbedo, H., Monden, Y., 1997. A parametric procedure for multcriterion sequence scheduling for just-in-time mixed-model assembly lines. International Journal of Production Research 35, 2543–2564.

Bautista, J., Cano, J., 2008. Minimizing work overload in mixed-model assembly lines. International Journal of Production Economics 112 (1), 177–191.

Bautista, J., Cano, A., 2011. Solving mixed model sequencing problem in assembly lines with serial workstations with work overload minimisation and interruption rules. European Journal of Operational Research 210 (3), 495–513.

Bautista, J., Cano, A., Alfaro, R., 2012. Modeling and solving a variant of the mixed-model sequencing problem with work overload minimisation and regularity constraints: an application in Nissan's Barcelona plant. Expert Systems with Applications 39 (12), 11001–11010.

Bautista, J., Companys, R., Corominas, A., 1996. Heuristics and exact algorithms for solving the Monden problem. European Journal of Operational Research 88, 101-113.

Bolat, A., 1997. Stochastic procedures for scheduling minimum job sets on mixed model assembly lines. Journal of Operational Research Society 48, 490-501.

Bolat, A., Yano, C., 1992a. A surrogate objective for utility work in paced assembly lines. Production Planning & Control 3, 406-412.

Bolat, A., Yano, C., 1992b. Scheduling algorithms to minimize utility work at a single station on paced assembly line. Production Planning and Control 3/4, 393–405.

Boysen, N., Fliedner, M., 2006. Comments on "Solving real car sequencing problems with ant colony optimization. European Journal of Operational Research 182, 466-468.

Boysen, N., Fliedner, M., Scholl, A., 2009. Sequencing mixed-model assembly lines: survey, classification and model critique. European Journal of Operational Research 192 (2), 349–373.

Golle, U., Boysen, N., Rothlauf, F., 2010. Analysis and design of sequencing rules for car sequencing. European Journal of Operational Research 206, 579-585.

Kim, S., Jeong, B., 2007. Product sequencing problem in Mixed-Model Assembly Line to minimize unfinished works. Computers & Industrial Engineering 53, 206-214.

Korkmazel, T., Meral, S., 2001. Bicriteria sequencing methods for the mixed-model assembly line in just-in-time production systems. European Journal of Operational Research 131, 188-207.

Kotani, S., Ito, T., Ohno, K., 2004. Sequencing problem for a mixed-model assembly line in the Toyota production system. International Journal of Production Research 42, 4955-4974.

Lesert, A., Alpan, G., Frein, Y., Noiré, S., 2011. Definition of spacing constraints for the car sequencing problem. International Journal of Production Research 49 (4), 963 – 994.

Miltenburg, J., 1989. Level schedules for mixed-model assembly lines in just-in-time productions systems. Management Science 35, 192-207.

Monden, Y., 1983. Toyota Production System. Industrial Engineering and Management Press, Institute of Industrial Engineers, Atlanta, GA.

Mosadegh, H., Zandieh, M., Fatemi Ghomi, S., 2012. Simultaneous solving of balancing and sequencing problems with station-dependent assembly times for mixed-model assembly lines, Applied Soft Computing 12, 1359-1370.

Scholl, A., Klein, R., Domschke, W., 1998. Pattern based vocabulary building for effectively sequencing mixed-model assembly lines. Journal of Heuristics 4, 359-381.

Xiaobo, Z., Ohno, K., 1997. Algorithms for sequencing mixed models on an assembly line in a sit production system. Computers and Industrial Engineering 32, 47–56.

Xiaobo, Z., Ohno, K., 2000. Properties of a sequencing problem for a mixed model assembly line with conveyor stoppages. European Journal of Operational Research 124, 560-570.

Yano, C.A., Bolat, A., 1989 Survey, development and applications of algorithms for sequencing paced assembly lines. Journal of Manufacturing and Operations Management 2, 172-198.

Yano, C.A., Rachamadugu, R., 1991. Sequencing to minimize work overload in assembly lines with product options. Management Science 37, 572-586.